

# **EZFILES**

# Table of Contents

Preface .....	4
Introduction .....	5
Directory Structure and Properties .....	6
Compaq Clusters (Open) .....	6
File-System Hierarchy .....	6
Directory Properties and Limits .....	8
Compaq Clusters (SCF) .....	9
SCF File-System Hierarchy .....	9
SCF Directory Properties and Limits .....	9
IBM SP .....	11
Pentium Linux Clusters .....	13
DFS and NFS Compared .....	14
NFS (Network File System) .....	14
DFS (Distributed File System) .....	15
Common Home Directories Summarized .....	17
File Management Techniques .....	20
Typical Code-Development Scenario .....	20
Transfer Rates for Mounted File Systems .....	21
File-System Usage Comparison .....	22
File-Sharing Alternatives Compared .....	23
Backup Policy Summary .....	24
Search Paths .....	25
Compaq-Cluster Search Path .....	25
Permissions for Files and Directories .....	27
Kinds of Permission .....	27
How to Discover Permissions .....	27
How to Change Permissions (CHMOD) .....	29
Absolute (Octal) Form of CHMOD .....	29
Symbolic Form of CHMOD .....	30
Citizenship and Permissions (SCF) .....	31
Top-Level World Permissions Disabled .....	32
File-Management Tools .....	33
GIVE .....	34
How to Run GIVE .....	34
GIVE Options .....	35
GIVE Examples .....	35
TAKE .....	37
How to Run TAKE .....	37
TAKE Options .....	38
TAKE Examples .....	38
QUOTA .....	39
How to Run QUOTA .....	39
QUOTA Options .....	39

QUOTA Examples .....	40
LIMIT .....	41
How to Run LIMIT .....	41
LIMIT Options .....	41
LIMIT Examples .....	41
DU .....	42
How to Run DU .....	42
DU Options .....	42
DU Examples .....	42
DF .....	44
How to Run DF .....	44
DF Options .....	44
DF Examples .....	45
MOLE .....	46
How to Run MOLE .....	46
MOLE Options .....	46
MOLE Examples .....	47
HTAR .....	49
Tools for Obsolete File Types .....	50
Using Groups .....	51
Disclaimer .....	54
Keyword Index .....	55
Alphabetical List of Keywords .....	57
Date and Revisions .....	59

# Preface

- Scope: EZFILES explains the hierarchical structure of directories and file systems on LC computers, the properties and limits of those directories, the role of the common home directory, and the default search paths. It also introduces some file-management software tools that monitor and report on important system features (such as quotas), and other tools that manipulate file permissions, transfer files, or perform basic file-handling tasks. Tools (such as GIVE and MOLE) and file features (such as citizenship groups) unique to the LC computing environment receive special attention in EZFILES.
- Availability: When the programs described here are limited by machine, those limits are included in their explanation. Otherwise, they run under any LC UNIX system.
- Consultant: For help contact the LC customer service and support hotline at 925-422-4531 (open e-mail: [lc-hotline@llnl.gov](mailto:lc-hotline@llnl.gov), secure e-mail: [hotline@pop.llnl.gov](mailto:hotline@pop.llnl.gov)).
- Printing: The print file for this document can be found at:
- on the OCF: <http://www.llnl.gov/LCdocs/ezfiles/ezfiles.pdf>  
on the SCF: [https://lc.llnl.gov/LCdocs/ezfiles/ezfiles\\_scf.pdf](https://lc.llnl.gov/LCdocs/ezfiles/ezfiles_scf.pdf)

# Introduction

This manual provides a basic guide to effectively managing your files and directories on LC computers. Its goal is to include the most important and relevant tools, techniques, and features, while making them easy to find and learn by avoiding more esoteric features or comprehensive descriptions of every possible feature.

EZFILES begins with a comparative look at how files and directories are organized on LC machines, with an emphasis on the important differences between each machine's home and work directories. We then look at the file-management implications of sharing the home directories across (most) open machines at LC. Included also are suggested techniques for working effectively in the public directory structure, a summary of which directories are backed up automatically and which are not, and a concise explanation of the default search path on different systems. EZFILES then introduces CHMOD to manage file permissions, and offers basic instructions for between-user file exchange (with GIVE and TAKE), home-directory and /nfs/tmp\* quota monitoring (with QUOTA), and other related file-management tasks. EZFILES gives special attention to file-handling tools unique to LC (not found in other UNIX environments), by explaining some (e.g., MOLE) and by linking to the local user manuals that explain others (e.g., HTAR).

Some UNIX file-management tools behave differently under the Linux version of UNIX or have extra options for added file-handling tasks. Consult LC's Linux Differences (URL: <http://www.llnl.gov/LCdocs/linux>) user guide for alerts about and details on these Linux-only features. For the (future) Linux-only parallel file system, see the manual called CHAOS: Linux from Livermore. (URL: <http://www.llnl.gov/LCdocs/chaos>)

# Directory Structure and Properties

This section shows how the important public directories (and their underlying file systems) are organized on the prime LC computers, and compares the home and work file environments on each computing system. The diagrams and tables make it easy to see the differences between different kinds of computer, or between computers of the same kind but differently configured on the open and secure networks.

## Compaq Clusters (Open)

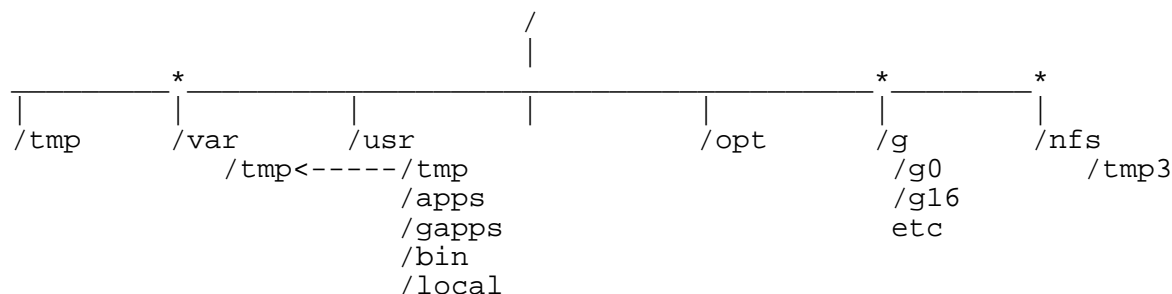
On the open network (OCF), LC provides several clusters of Compaq (formerly DEC) machines on a "general availability" basis to support "Multiprogrammatic and Institutional Computing" (M&IC):

- GPS--a cluster of 49 nodes (4 CPUs/node, 2 Gbyte of memory on interactive nodes but up to 32 Gbyte on batch nodes) called GPS01, etc., and running Tru64 UNIX.
- TC2K--a cluster of 128 nodes (4 CPUs and 2 Gbyte of memory per node) called TC2K0, etc., and running Tru64 UNIX.

All these computers have the same basic file-system hierarchy and public-directory properties, described in the subsections that follow, except where differences (usually pertaining to Linux) are explicitly noted.

## File-System Hierarchy

Each machine in the open Compaq clusters has a public directory structure of the form shown here (key directories are marked \*):



where

**/tmp** is a small directory for system use only (very transitory files). On all Linux clusters *only*, /tmp is just a link to /var/tmp.

**/usr/tmp** is simply a convenient link pointing to /var/tmp (so also is /usr/var/tmp).

**/var/tmp** is the intended main working directory. You should create a subdirectory here (or in /usr/tmp if you prefer) and conduct most computational work in it. See the next section for a comparative look at the properties of /var/tmp (keyword: [dec-properties](#) (page 8)), and see the section on file management techniques (keyword: [work-steps](#) (page 20)) for a summary of sound work practices.

/nfs/tmp3	is an alternative working directory with properties similar to /var/tmp (keyword: <u>dec-properties</u> (page 8)). Access is slower than for /var/tmp, but /nfs/tmp3 is global, that is, shared among (NFS-mounted on) all open Livermore Computing production machines to eliminate the need for between-machine file transfers and to simplify batch-job preparations. You should create your own subdirectory here for your computational work (rather than run big jobs in your home directory). Whenever this directory becomes 90% full, files are purged by age until it becomes no more than 70% full. (Former /nfs subdirectories /tmp0, /tmp1, and /tmp2 no longer exist.)
/g	is a file system of globally available ("common") home directories on highly reliable RAID disks NFS-mounted on each open production machine. Your child subdirectory here (/g/gnn/yourname) is your default arrival directory and contains your startup and run-control dot files, but it is limited in size. See the next section (keyword: <u>dec-properties</u> (page 8)) for a comparison of home and /var/tmp properties, and see the section on common home directories (keyword: <u>common-home</u> (page 17)) for more details on the default contents of this directory.
/opt	contains some DCE support tools (such as DCECP and KINIT) on the GPS and TC2K clusters. On LX, /opt is empty (but it contains the Intel compilers on ILX).
/usr/apps	is a link to /usr/gapps/\$SYS_TYPE (beginning in May, 2001). See /usr/gapps below.
/usr/gapps	is a file system of globally available ("common") code-management directories on RAID disks NFS-mounted on each machine, just as with the /g common home directories. The /usr/gapps directories contain some important <i>noncommercial</i> shared application codes and tools, such as BASIS, YORICK, JEEP, SILO, and TART, segregated into subdirectories by operating system. See the <u>Common Home Reference Manual</u> (URL: <a href="http://www.llnl.gov/LCdocs/chome">http://www.llnl.gov/LCdocs/chome</a> ) for details on the complex internal structure of /usr/gapps.
/usr/bin	contains most standard UNIX software tools along with the C and Fortran compilers.
/usr/local	contains some important <i>commercial</i> shared tools, such as TOTALVIEW, MATLAB, MATHEMATICA, PVM, GUIDE, and ASSURE. These tools may not be available on Linux machines; see the <u>Linux Differences</u> (URL: <a href="http://www.llnl.gov/LCdocs/linux">http://www.llnl.gov/LCdocs/linux</a> ) guide for an availability list.

## Directory Properties and Limits

Because of their different roles, the home (/g) and work (/var/tmp, /nfs/tmp3) directories on the open Compaq cluster machines have different properties, and understanding these differences can help you use each directory in the most appropriate way:

	/g/gnn/uname	/var/tmp	/nfs/tmp3
Role:	home directory	working directory	working directory
Aliases:	~ \$HOME	/usr/tmp (*)	[none]
Status:	NFS mounted, so shared	local to each machine	NFS mounted, so shared
Shared across machines?	yes	no	yes
Quotas?	yes	no	yes
...On file size:	16 Gbyte/user	120 Gbyte/machine (60 Gbyte on TC2K)	100 Gbyte/user (7.2 Tbyte total)
...On file count(+):	no limit	no limit	1,000,000/user
Purge?	no	yes	yes, starts when 90% full
Files vulnerable:	never	after 3 days	after 10 days
Automatic backup?	yes, 4 copies, every half day	no, use storage	no, use storage

(\*)WARNING: on the open Compaq machines, the environment variable WRK evaluates to /g/ggroup/yourname rather than to /var/tmp, that is, to your home rather than to your work directory. So do NOT rely on the command `cd $WRK` to move to your work directory (for example, in batch scripts from elsewhere).

(+)The limit here is strictly speaking on *inodes*. An inode is the "index node" by which UNIX file systems keep track of the (often scattered) disk blocks that comprise each file. Since inode size is fixed, a large file may require more than one inode to list all of its disk blocks. Hence, users with large files may find that slightly less than 1,000,000 files are allowed.

A later [section](#) (page 9) shows the same comparative table for SCF machines. See the [Common Home section](#) (page 17) below for a way that your application code can test for the presence of globally mounted directories on any LC machine where it runs.



## Compaq Clusters (SCF)

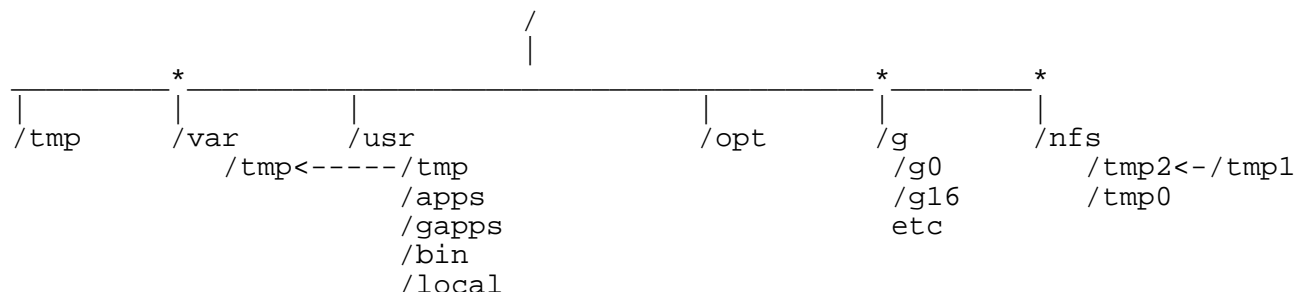
On the secure (SCF) network, LC provides two nonASCII clusters of Compaq (formerly called DEC) machines on a "general availability" basis to support scientific computing:

- SC--a cluster of 40 nodes (4 CPUs and 4 GByte of memory per node for 32 nodes; 8 nodes have 32 GByte of memory) called SC1 etc., and running Tru64 UNIX.
- ICF--a cluster of 12 nodes (3 CPUs and 1 GByte of memory per node) running Tru64 UNIX. (See also the "Pentium Linux Clusters" section, below (page 13).)

These machines have mostly the same file systems and public-directory properties as the open clusters described above (page 6). This section therefore points out only the important differences you will encounter on the SCF Compaqs, and refers you to the open Compaq section above for relevant general background on file systems and directory properties.

## SCF File-System Hierarchy

Each machine in the secure Compaq clusters has a public directory structure of the form shown here (key directories are marked \*):



which is identical to the open Compaq directory structure (page 6) EXCEPT that:

(1) the shared (NFS-mounted) work directories under /nfs are called /tmp0 and /tmp2 (where /nfs/tmp1 became just a link to /tmp2 in July, 2001), and

(2) your secure common home (page 17) directory involves /g and its children if your SCF account was first created after October 1, 1999 (/g here behaves just like /g on open machines). But if your SCF account was first created BEFORE October 1, 1999, then your default home directory was once a child of /u and was migrated to /g late in 2001. Where the /u files landed depends on whether you also had a /g directory: those already with a /g retained their old files in /g/gnn/yourname/forest; those without /g retained their files in a new /g itself. To avoid file-name collisions, these /forest subdirectories of /g persist (until you personally empty them), even though the SCF Forest cluster of machines for which they were named itself disappeared from SCF in October, 2002. See the Common Home Reference Manual (URL: <http://www.llnl.gov/LCdocs/chome>) for full migration details.

## SCF Directory Properties and Limits

The home and work (/var/tmp) directories on the secure Compaq cluster machines differ just as they do on the open Compaqs, with a few important exceptions marked (+) in this chart:

	<b>/g/gnn/uname</b>	<b>/var/tmp</b>	<b>/nfs/tmp0, /tmp2</b>
Role:	home directory	working directory	working directory
Aliases:	~ \$HOME	/usr/tmp (*)	/nfs/tmp1
Status:	NFS mounted	local to each machine	NFS mounted
Shared across machines?	yes	no	yes
Quotas?	yes	no	yes
...On file size:	2 Gbyte total	53 Gbyte/node(+)	100 Gbyte/user
...On file count(\$):	no limit	no limit	1,000,000/user
Purge?	no	yes	yes, starts when 90% full
Files vulnerable:	never	after 10 days(+)	after 10 days
Backup?	yes, 4 copies, every half day	no, use storage	no, use storage

(+)Indicates an important DIFFERENCE between SCF directories and the corresponding open directory properties. SC cluster nodes SC21 to SC28 have only 24 Gbyte/node of /var/tmp space.

(\*)WARNING: on the SCF Compaqs, the environment variable WRK is not defined. So do NOT rely on the command `cd $WRK` to move to your work directory (for example, in batch scripts from elsewhere).

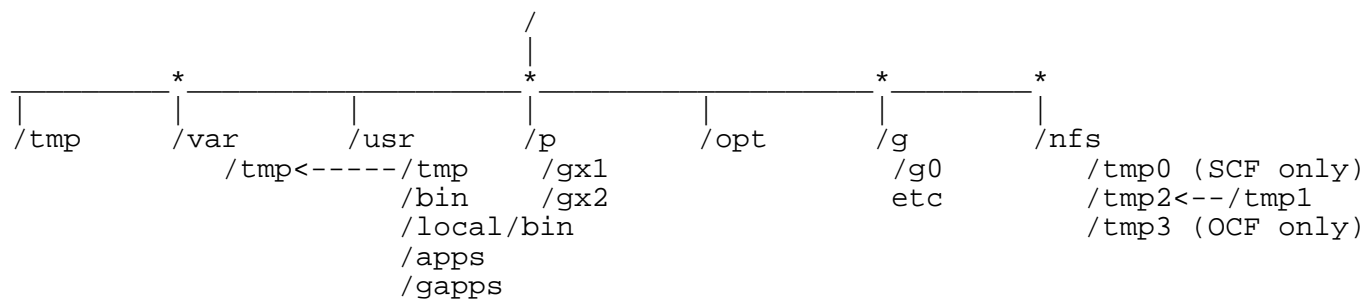
(\$)  
The limit here is strictly speaking on *inodes*. An inode is the "index node" by which UNIX file systems keep track of the (often scattered) disk blocks that comprise each file. Since inode size is fixed, a large file may require more than one inode to list all of its disk blocks. Hence, users with large files may find that slightly less than 1,000,000 files are allowed.

An earlier [section](#) (page 8) shows the same comparative table for open-network (OCF) machines. See the Common Home [section](#) (page 17) below for a way that your application code can test for the presence of globally mounted directories on any LC machine where it runs.

# IBM SP

LC's IBM SP machines are massively parallel sets of RS-6000 processors configured for ASCI production computing. Blue and Frost are currently open-network machines, while White, Ice, and the Blue-Pacific S machine are secure (SCF).

Each massively parallel IBM SP machine has a public directory structure of the form shown here (key directories are marked \*):



This structure offers the same directories serving just the same roles as on the open Compaq (page 6) cluster (described above), with only these differences:

- /opt contains only a link to the location for the DCE tools.
- Each IBM machine has two "General Parallel File Systems" (GPFS) intended specifically for large-file parallel I/O. Each GPFS name has the form /p/gx1 and /p/gx2, where *x* is a letter that designates the host computer or sector (so /p/gb1 is on Blue, while /p/gs1 is on the S sector of SKY). On the IBMs you should continue to use the NFS-mounted directories (such as your common home directory (page 17) and /nfs/tmp*n*) for source files, executables, and other relatively small files (for better service and more efficient resource utilization). The GPFS directories have slow metadata operations and a large block size, so they are designed primarily for handling large data files that need high-speed parallel I/O access from all SP nodes. See the I/O Guide for LC (URL: <http://www.llnl.gov/LCdocs/ioguide>) for locally relevant GPFS technical information.
- Both OCF and SCF IBM machines have globally mounted /nfs/tmp*n* work-directory file systems (but on SCF /tmp1 is merely a link to /tmp2, starting in July, 2001). Only OCF machines have /nfs/tmp3. Naturally, the files/user limit of 1,000,000 applies to all /nfs/tmp systems here as well as on the Compaqs.

This table summarizes the properties and limits of the (pair of) GPFS file systems available on each IBM SP (for easy comparison with the other directories similarly tabulated for the open Compaq (page 6) machines):

	<b>GPFS (/p/gx1, /pgx2)</b>
Role:	fast parallel I/O
Aliases:	[none]
Status:	not NFS mounted, IBM SPs only
Shared across nodes?	yes
Quotas?	no
On file size:	18.5 Tbyte total (OCF), 60 Tbyte total (SCF)
On file count:	no limit
Purge?	yes, summarized below(+)
Files vulnerable:	voluntary cleanup
Automatic backup?	no, use storage

(+)Using GPFS effectively and appropriately requires storing your files so that you avoid needlessly clogging the file system, and especially so that you avoid losing valuable data to the GPFS file purge.

- **Threshold:**  
LC purges GPFS when and only when usage exceeds 80%.  
Purging of files continues (oldest files first) until the file system is no more than 70% full (see schedule below).
- **Scope:**  
All GPFS files at least 13 weeks old (= 3 months) are eligible to be purged if the purge threshold is reached, but...  
All GPFS files 100 Kbyte or less in size are exempt from the purge regardless of their age.
- **Schedule:**  
If usage reaches the purge threshold during any month, then LC will start purging eligible GPFS files on the third Tuesday of that month (and continue until usage sinks to 70%).  
On the first Tuesday of every month, pre-purge logs are available for every user in a directory called

`/p/gxx/purgelogs/username`

(to help you anticipate which files are vulnerable for purge that month).

## Pentium Linux Clusters

On the secure network (SCF), LC provides two ASCI-funded "production capacity resource" (or PCR) clusters based entirely on Intel Pentium chips for general scientific computing. A separate ACE cluster provides similar resources but *without* any high-speed interconnect (switch). The login nodes use Pentium 3 chips, while the production compute nodes feature Pentium 4 chips with a 1.7 GHz speed:

- Adelie--a cluster of 128 Pentium nodes (2 CPUs and 2 Gbyte of memory per node) called adelie0 through adelie127. All run Red Hat Linux as modified at LLNL ([CHAOS](http://www.llnl.gov/LCdocs/chaos) (URL: <http://www.llnl.gov/LCdocs/chaos>)).
- Emperor--a cluster of 128 Pentium nodes (2 CPUs and 2 Gbyte of memory per node) called emperor0 through emperor127. All also run Red Hat Linux as modified at LLNL ([CHAOS](http://www.llnl.gov/LCdocs/chaos) (URL: <http://www.llnl.gov/LCdocs/chaos>)).
- ACE--a cluster of 64 Pentium nodes (2 CPUs and 4 Gbyte of memory per node) called ace1 through ace64. All run Red Hat Linux as modified at LLNL ([CHAOS](http://www.llnl.gov/LCdocs/chaos)).

These three Linux clusters have the same file systems and the same public-directory properties as the SCF Compaq clusters described in a [previous section](#) (page 9), except for the amount of local disk space. The per-node /var/tmp disk space on both Adelie and Emperor is 68 Gbyte, which is about 25% more than on most SC nodes (/var/tmp is 43.8 Gbyte on ACE nodes). So if you can handle the compiler differences (see the [Linux Differences](#) (URL: <http://www.llnl.gov/LCdocs/linux>) guide for details, suggestions, and special directories), then the Adelie and Emperor clusters will accommodate parallel codes with large disk-space needs more easily than will other SCF computing resources. Because ACE lacks a switch it is less suited for many-node parallel runs.

Remember also that while /dfs is shared among IBM and Tru64 (such as SC) nodes on SCF, no Linux nodes (neither ACE nor PCR nodes) have /dfs mounted on them.

On the open network (OCF), LC provides two large Pentium Linux clusters for general use:

- ILX--a cluster of 67 Intel Pentium nodes (2 CPUs and 4 Gbyte of memory per node) called ILX1 through ILX67. All run Red Hat Linux as modified at LLNL ([CHAOS 1.2](http://www.llnl.gov/LCdocs/chaos) (URL: <http://www.llnl.gov/LCdocs/chaos>)).
- MCR--a cluster of 1152 Intel Pentium nodes (2 CPUs and 4 Gbyte of memory per node) called MCR1 through MCR1152. All run Red Hat Linux as modified at LLNL ([CHAOS 1.2](http://www.llnl.gov/LCdocs/chaos) (URL: <http://www.llnl.gov/LCdocs/chaos>)).

Directories on these machines are the same as on the open [Compaq clusters](#) (page 6), except that /tmp is just a link to /var/tmp and /opt contains the Intel compilers.

## DFS and NFS Compared

LC provides its common home directories (/g and its shared temporary disk space (/nfs/tmp0 and /nfs/tmp2 (SCF), /nfs/tmp3 (OCF)) by using the Network File System (NFS) as implemented by a vendor called Network Appliance. Separately, and for more specialized purposes, LC provides other shared disk space by using a pilot implementation of the Distributed File System (DFS). Thus although they have similarities, at LC NFS and DFS are parallel projects with different detailed features deployed to meet different needs.

### NFS (Network File System)

NFS (Network File System), in various implementations, is a widely used service that allows disks (file systems) to appear local to multiple machines so that overtly copying or moving files between those machines (such as with FTP) is no longer necessary. LC uses NFS to support the common home (page 17) (/g) and shared temporary (/nfs/tmp*n*) directories that appear local to all its open computers (separate NFS installations provide similar service for several separate clusters of secure machines as well). In 2003, LC significantly upgraded its open NFS capacity to increase the reliability of its shared file systems and to expand the available shared disk space to 7.2 TByte (divided by quota (page 39) among users).

NFS has the potential for some security weaknesses, however, that are of concern if its use is not limited to a LAN environment where overall network and system security are well managed. Most NFS compromises are the result of configuration errors. LC is cognizant of the various types of configuration errors and is extremely careful regarding the configuration of the various NFS servers. Configurations are reviewed by senior technical staff to ensure their correctness.

The foremost intrinsic NFS vulnerability relates to an object called a file handle. If discovered or otherwise captured, a file handle for a particular file system allows anyone from any platform to easily access any data on that file system. The challenge is to capture a file handle. Three known methods to do this are discussed below:

- **Stealing.** In the past, various versions of UNIX operating systems allowed file handles to be stolen either from kernel memory or from tricking system daemons such as portmap to disclose it. LC has attempted to steal file handles using well-known "hacker" tools from its current production systems and has been unable to do so. The server operating system (NetApp Data ONTAP) appears to restrict access to file handles appropriately. In addition, the client operating systems in use on LC systems (e.g., Digital UNIX, AIX, IRIX, and Solaris) appear to properly safeguard file handles.
- **Snooping.** Snooping a file handle from the network generally requires the compromise of a system (i.e., gaining root privilege). By policy, access to file systems from LC NFS servers are allowed only to LC managed and root-controlled systems. Further, snooping in general (e.g., from a PC) is not at all easy since LC makes extensive use of switch-based networks. NFS traffic to and from the servers from the production hosts are through private Fast Ethernet (switched) networks or through switched FDDI links. Snooping a file handle is extremely unlikely.
- **Masquerading.** Masquerading involves one system taking over the identity (i.e., IP address) of another system, typically when the other system is turned off. LC systems are operated on a 24-hour basis with a 24-hour operations staff; the possibility of masquerading is extremely remote, and NFS traffic is blocked from entering or leaving the OCF network by router filters.

In summary, as long as NFS traffic is kept within a controlled LAN environment and the servers are properly configured, it is unlikely NFS can be compromised. There is probably a greater likelihood of compromising root on a given system, which LC does all it can to guard against.

## **DFS (Distributed File System)**

To provide file sharing in an "enterprise" or WAN environment where security domains and administrative control areas are crossed, LC has been working on implementing DFS (Distributed File Service). Whereas NFS (page 14) is currently not considered a safe technology for sharing files outside a LAN environment, DFS is safe since it relies on the Distributed Computing Environment (DCE) security component to authenticate a user. DFS provides a uniform global name space and supports fine grain access control to files and directories using Access Control Lists (ACL).

The LC DFS pilot environment currently has a disk capacity of 250 GB. On the open network, DFS is accessible from the GPS and TC2K clusters and from the IBM ASCI systems. On the secure network, DFS is mounted on the SC cluster as well as the IBM (ASCI) systems. DFS is *not* available on any LC Linux system. A DFS client for the visualization server is currently being tested. DFS is a maturing technology that is stable throughout most of the LC environment, although problems exist on some platforms, most notably Digital UNIX. LC continues to work closely with vendors such as Compaq to resolve these problems, and continues to test all the client implementations of interest. The PSUB utility to submit batch jobs has already been revised to get the authentication needed to let batch jobs access DFS files.

LC's initial use for DFS is targeted at projects or groups interested in having a shared file space extended to their desktop or shared between the DOE laboratories. Extension to UNIX and Windows NT desktops has already been accomplished for members of an atmospheric sciences research group at LLNL. Another targeted use is to address concerns for protection of export controlled software and the protection of intellectual property. These projects use DFS for establishing the proper modes of access, by granting full access permissions for application developers, limiting access of application users, and denying access to others. LC has also been working on integrating Web access with DFS; this brings the security features of DFS to the Web and the Web data is location independent.

Beginning in February, 2001, LC uses the standard DFS file-permissions mechanism (the "access control list" or ACL) to disable access to TOP-level DFS project directories for some users as part of its security policy. For each file and directory in DFS, an access control list specifies for the following sets of users (called "ACL entries")

user_obj	the owner
group_obj	the group assigned to this file
other_obj	any user in the local DCE cell
foreign_other	any user from a DCE cell not locally administered (e.g., Sandia and LANL are "foreign" to LLNL's cell)
any_other	any user in the world

the access permissions (read, write, etc.) that those users have. Within DFS this ACL approach overrides the usual UNIX permissions.

DFS project directories have names of the form */dfs/proj/codename*. The access restrictions on these directories now work as follows:

- (1) The */dfs/proj* directory now has no *foreign\_other* entry at all in its ACL.
- (2) Each of the 21 individual */dfs/proj/codename* directories (such as */dfs/proj/ale3d*) now has no *foreign\_other* and no *any\_other* entry in its ACL.
- (3) The *other\_obj* entry remains in the ACL for each */dfs/proj/codename* directory, but no permissions (000) are assigned to it.

Permissions remain unchanged for all files and directories below the */dfs/proj/codename* level, but if your access to them depends on your membership in the *foreign\_other* or *any\_other* sets of users, then your ability to read (or even to list) these low-level files may be disabled. To display the current ACL assigned to any file or directory within DFS, use this execute line:

```
acl_edit pathname -l
```

For current technical details on the tools and techniques needed to manage files under DFS at LC, consult the draft local documentation available online at this URL (DCE password required):

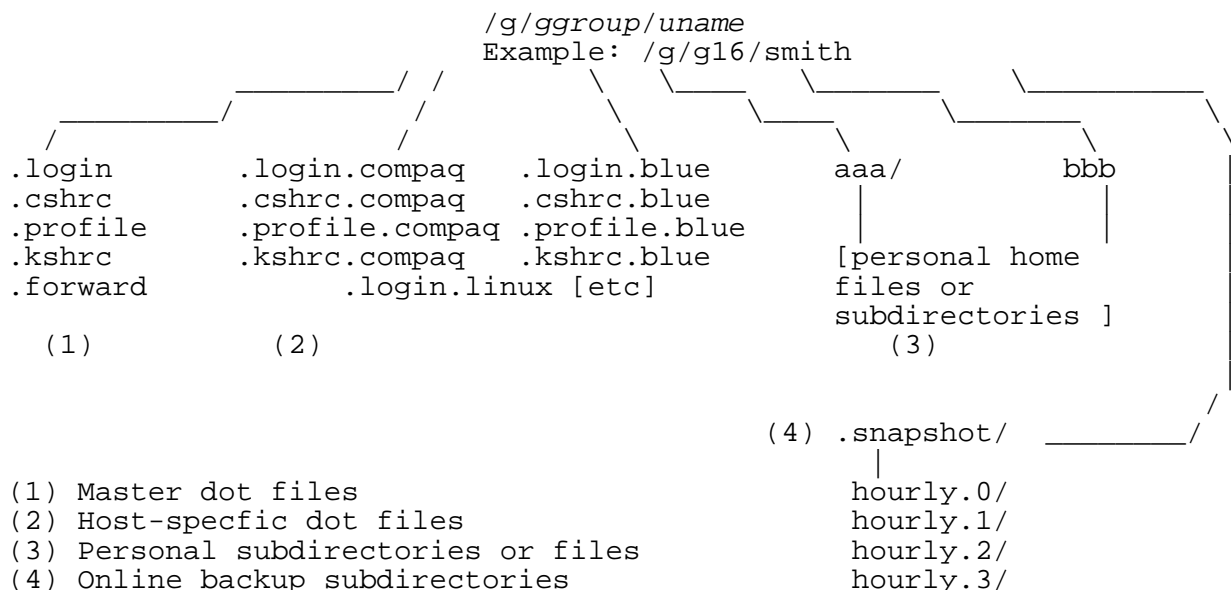
```
Open: https://lc.llnl.gov/computing/docs/dfsintro4.html
SCF:  [not yet available]
```



# Common Home Directories Summarized

All machines in the open clusters (GPS, TC2K, ILX, MCR) and all nodes on the open ASCII Blue and Frost IBM computers share home directories that reside on global file system /g, NFS mounted from several dedicated servers. Likewise, all machines in the production secure clusters (SC, Emperor, Adelie) and all nodes on the secure ASCII White, Ice, and Blue-Pacific S IBM computers share their own secure home directories on a global SCF/g file system. This too is NFS mounted across machines from multiple servers. (Legacy SCF users whose accounts were first created before October 1, 1999, have had their SCF home directory moved from the /u to the /g file system, as explained in the Common Home Reference Manual. (URL: <http://www.llnl.gov/LCdocs/chome>))

This "common home" arrangement makes keeping redundant home files (and doing redundant updates) on multiple machines unnecessary, and it allows the same pathname and directory structure for home to be shared on every host involved:



where

*ggroup* is an administrative subdirectory different from the UNIX groups that LS or GROUPS reports. On OCF, it is g0 for LC staff and g2 or above for other users; on SCF, it is g5 for LC staff and g10 or above for other users, and where

*uname* is your alphabetic login account name (not the numeral *uid* by which your file and block quotas are reported).

## (1) Master dot files

are your basic startup and run-control files (copied from /gadmin/etc/skell). They detect the machine type you are on (and the operating system you are under) by evaluating the HOST\_GRP and SYS\_TYPE environment variables, and they then invoke the appropriate machine- (or system-)specific dot files (shown here at (2)).

Only customizations intended for ALL machine types and operating systems should go into these master dot files.

## (2) Host-specific dot files

are the place for your customizations that apply only to individual hosts or to specific operating systems. "Host-group" names as suffixes (e.g., COMPAQ for all the open Compaq machines, BLUE for all open IBM SP nodes) simplify sharing these customizations among like machines (only).

## (3) Personal files and subdirectories

are whatever other files you want to have in your home directory, up to your quota, organized as you wish under /g/ggroup/uname. These files will appear to reside on every machine that shares the common home directory.

## (4) Online backup subdirectories

contain four complete (but read-only) backup copies of every file and subdirectory in your common home directory, made automatically at noon and 7 p.m. every day. The most recent backup resides in .snapshot/hourly.0, with each earlier backup in a correspondingly numbered hourly.*n* directory. See the Backup section in the Common Home Reference Manual (URL: <http://www.llnl.gov/LCdocs/chome>) for technical details.

## CAUTIONS.

Several cautions apply to the common home directories:

First, common home directories all have disk-space quotas: 16 Gbyte for general users (8 Gbyte for LC staff). Note also that the output from quota-monitoring tools varies somewhat from one brand of computer to another. Consult the QUOTA (page 39) section below for a comparison of disk-usage reporting formats.

Second, your common home directory by default allows access only to you (as the owner). You can widen this access (by running CHMOD, whose instructions appear in a later section, keyword: chmod (page 29)). But remember that the changed permissions will apply to ALL hosts sharing this directory, not just to the one host on which you make the change, as would usually be the case. Because of the danger of lost data or hidden changes to your login files, WORLD write access to your top-level common home directory (i.e., to /g/ggroup/uname) is NOT allowed on open machines. And world or even group write access to /g/group/uname disables SSH as well. You can enable world or group write access to your home subdirectories if you wish, but this too is risky. Consider using GIVE (page 34) or TAKE (page 37) instead.

Third, you cannot checkpoint (for restart) any job whose files reside on NFS-mounted disks. Since your common home directory is NFS-mounted, any batch job you run on a common-home machine that spawns a shell will access its dot files on those disks, and hence, that job will not checkpoint.

For additional technical details on the behavior of common home directories (and their quotas), on host-group names, and for advice if problems arise with specific software, consult the Common Home Reference Manual (URL: <http://www.llnl.gov/LCdocs/chome>) on the LC documentation web servers.

## TEST.

If you have a single version of an application code (or utility) that you run on diverse machines, you may need to test for the presence or absence of LC common home directories to decide about where to locate (initialization or output) code-related files. To facilitate this test, each LC machine offers a system file called `/etc/home.config`, in which one line contains the keyword "FILE\_SYS" (uppercase) followed by one of four configuration values (shown below, mixed case). Your code can reveal the content of this line, for use in conditional tests, by executing

```
grep FILE_SYS /etc/home.config
```

The four possible configuration values that this might return are:

global	means that all LC global file systems, including the common home directories of <i>all</i> users, are available.
LC_only	means that only LC staff common home directories are available; home directories for general users are <i>not</i> mounted here (e.g., LUCY).
green	means that only unrestricted-network (local) home directories are available (e.g., for "foreign national" users).
pl1	means that this SCF system at security level PL1 has <i>no</i> global (including common home) file systems available.

# File Management Techniques

## Typical Code-Development Scenario

This scenario suggests practical and appropriate steps for managing your files while you develop code on any open LC cluster. The same basic approach also applies while working on the nodes of the secure LC clusters:

(1) Log on. You will arrive in your home directory, which on the open clusters is the "common home" directory `/g/ggroup/uname` that all such machines share. Because it is not purged, your home directory is a good place for various background and support files. But because of its size quotas, developing large projects in your home directory is often impractical.

(2) Use `CD` to move to `/var/tmp` (or to `/nfs/tmp0` or `/nfs/tmp2` (SCF), or `/nfs/tmp3` (OCF)). This puts `/var/tmp`, which is NOT on your default search path, into your search path as the "current working directory" (`.`), a convenience for executing your own programs. Create a subdirectory (usually named after your logon id for clarity, such as `/var/tmp/jsmith`), and `CD` into it. This makes the relatively large size of `/var/tmp` (or `/nfs/tmpn`) available for your work without mingling your files with others.

(3) Create symbolic links from pointers in your subdirectory of `/var/tmp` (or `/nfs/tmpn`) to your code (and perhaps to other key files) in your home directory, in this way:

```
ln -s origfile pointname
ln -s ~/proj2 proj2.tmp
```

These links must be symbolic (`-s`) because your home directory, `/var/tmp`, and `/nfs/tmpn` do not reside on the same file system.

(4) Edit, compile, test, and debug your code in your subdirectory of `/var/tmp` or `/nfs/tmpn`. Because of the symbolic links, your changes and updates end up mapped safely back to your home directory as a side effect of your work in `/var/tmp`. And because `/var/tmp` has no quotas, you enjoy much greater flexibility in using large data sets or making large output files. This approach also mitigates the occasional slowness of working on mounted file systems directly.

(5) Store your results and updates often. Use FTP (or, where available, NFT) and transfer copies to `storage.llnl.gov` (open or secure). Remember that `/var/tmp` (and `/nfs/tmpn`) on Livermore Computing production machines are NOT automatically backed up, and that all files older than a threshold age (now 3 days on the open clusters) can be purged with no possibility of recovery. With regular archival storage and good housekeeping you can delete unneeded files and leave more space for other users.

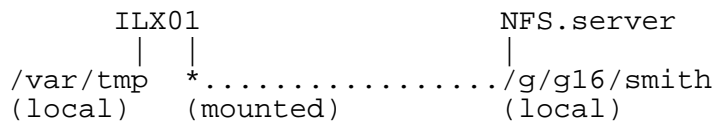
(6) There is no need to move files in your home directory (or in `/nfs/tmpn`) on Livermore Computing production machines between MACHINES because your home directory is shared among all open machines (on the open network) or among all secure machines (on the secure network). The other working directories (such as `/var/tmp`) are local to each machine, however, and you should use FTP (or, where available SCP or NFT) to move work-directory files from `/var/tmp` on one machine to `/var/tmp` on another (if you need to).

(7) On IBM machines only, the batch system automatically sets the environment variable `LOADL_STEP_ID` and then creates a job subdirectory called `/var/tmp/$LOADL_STEP_ID`. If you run

your batch job in that specific directory, then all of its files will automatically disappear when the job ends (so save them first!), guaranteeing that you will not hoard /var/tmp space on the IBM compute nodes.

## Transfer Rates for Mounted File Systems

To make files more readily available across machines, a file system local to one machine is sometimes "NFS-mounted" on (or "exported to") another machine. For example:



Here the home directory /g/g16/smith is really local to some NFS server machine, but it has been mounted on ILX01 (a Linux cluster node) and so it appears there as well, where you can use ordinary UNIX tools (LS, RM, CHMOD, etc.) to manipulate its files as if they were local to ILX01.

When TRANSFERRING large files between a mounted and local file system, however (for instance, between /g/g16/smith and /var/tmp on ILX01 here), efficiency issues can affect the best choice of transfer tool to use. Copying the files with CP is the most natural and simple approach, and is fine for small files (behind the scenes a remote copy is really taking place between machines). But for large or multiple files, using FTP for the transfer instead of CP, even though only one host seems to be involved, proves much more efficient.

To transfer a mounted file both FTP and CP must open the source file, read the file's bits, open a destination file, and write its bits. But the transmission rate for CP is severely limited by the NFS mounting mechanism, while FTP efficiently bundles the transmitted bits to take (better) advantage of the maximum bandwidth available. In the example above involving a file system cross mounted on two machines at LC, the typical (between-machine) transmission rate for CP is only 0.5 Mbyte/s but the typical rate for FTP is almost 5 Mbyte/s, ten times faster. So this significant difference justifies the extra setup that FTP requires if many or large files are being transferred. (Remember too that this difference applies ONLY to mounted file systems, and that CP has no disadvantage in speed and an obvious benefit in simplicity for ordinary file transfers done WITHIN a single file system.)

## File-System Usage Comparison

On LC production machines, different file systems (appearing as different top-level directories) have different features and hence different intended uses. This chart summarizes the most suitable use for each LC file system, to help you compute efficiently and avoid problems. Each itemized file system is also linked to its detailed technical description elsewhere in EZFILES (or in another manual).

<b>File System:</b>	<b>Most Suitable For:</b>
<u>Home directory</u>	
/g/gnn/uname	Small, permanent files shared among machines (dot files, source code)
<u>Local temporary directories</u>	
/tmp	Only for system use (under Linux, links to /var/tmp)
/usr/tmp	A link to /var/tmp
/var/tmp	Temporary storage of input or output local to each machine
<u>Global temporary directories</u>	
/nfs/tmp0 (SCF)	Same role as /nfs/tmp3
/nfs/tmp2 (SCF)	Same role as /nfs/tmp3
/nfs/tmp3 (OCF)	Large temporary storage for input or output shared among many machines
<u>General parallel file system</u>	
/p/gx1 (larger)	Large data files,
/p/gx2 (smaller)	fast parallel I/O shared among IBM SP nodes
<u>Storage directories</u>	
[ftp storage]	Long-term (archival) storage of any size or type of file

## File-Sharing Alternatives Compared

The standard UNIX technique for sharing access to files among several users is to enable (read, write, or execute) permissions (page 27) on those files (and their directory trees) for "world" or "other" users. On LC production machines, however, all top-level world permissions are automatically disabled (page 32) (set to 0) by monitoring software as a security policy. This effectively prohibits world-permission file sharing at LC. (An exemption requires specific approval of your LLNL Associate Director; contact the LC Hotline for details.)

Here are several alternative file-sharing techniques, each with its own strengths and weaknesses (and each linked to its execution details elsewhere in EZFILES):

- Run utilities GIVE (page 34) (to copy a file to another user) and TAKE (page 37) (to receive a GIVEN copy). This is well suited to sharing a few seldom-changed files with another specific user, but not to sharing large sets of files with many users, especially if the files change often. Also, the sharing users must all work on the same SCF or Linux/CHAOS machine; on OCF AIX or Tru64 machines you can GIVE files on one machine and TAKE them on another.
- Use file group (page 51) membership and permissions rather than world permissions. This is well suited to sharing large sets of files or whole subdirectories with a stable list of other users, and it allows across-machine sharing (if the files are in a globally mounted file system, such as the common home directories, and if the same user group exists on several machines). But an administratively approved form is required to create a group, and no LC user can belong to more than 32 groups at once.

One variation on file sharing by group involves enabling group permissions on file(s) parked in /usr/gapps (common to all production machines). This requires completing the special USR\_APPS form (because /usr/apps is just a link to /usr/gapps/\$SYS\_TYPE).

A second variation involves enabling group permissions on STORED files. Storage groups and online groups are not the same, however, and group assignment changes when a file is stored. See the EZSTORAGE (URL: <http://www.llnl.gov/LCdocs/ezstorage>) basic guide for detailed instructions on sharing stored files.

- Use DFS (page 15) (Distributed File Service) with an access control list. This allows each shared file to have its own fine-grained list of sharing users. But the files must reside on a (still-developmental) DFS server, whose DFS file-management utilities are exotic and fairly complex.

# Backup Policy Summary

Because some file systems contain valuable, frequently used information whose loss would be very disruptive, LC uses Legato Networker software to make backup copies of these file systems on tape. Full backup copies are made once each month, and incremental (changes-only) backups occur every night except Saturday and Sunday. LC copies over 400 Gbytes of unclassified data and over 90 Gbytes of classified data to backup tapes every month.

Other file systems, however, although heavily used for important work, are simply too large to allow practical backup copies to be made, or they are mounted on machines for which no appropriate backup software is commercially available. For files on these systems, no second copy automatically exists. Here your responsibility is to personally make storage copies of all crucial files in case you need to restore them on your own after disk problems.

This chart summarizes the backup status for each major file system on the LC production machines (those not listed are also NOT protected by backup):

File system	Backup status	
	Automatic backup On ALL machines	NO backup here
/g/gnn[ * ]	X	
/usr/local	X	
/usr/gapps[ * ]	X	
/usr/tmp		X
/var/tmp		X
/nfs/tmp0 (SCF)		X
/nfs/tmp2 (SCF)		X
/nfs/tmp3 (OCF)		X
/p/gx1 (GPFS)		X
/p/gx2 (GPFS)		X
/dfs	X	

[\*]See the Backup section in the [Common Home Reference Manual](http://www.llnl.gov/LCdocs/chome) (URL: <http://www.llnl.gov/LCdocs/chome>) for how to retrieve these "layered" backups.



# Search Paths

When you try to execute a program by typing its (simple) file name, your UNIX shell searches through the file structure looking for a file with that name to execute. The order in which it searches directories is specified by your search path, a colon-delimited ordered list of directories stored in the environment variable PATH (all uppercase). If you use a program's absolute pathname, the shell ignores your search path. Search paths often differ drastically from one brand of computer to another because of operating-system differences. You can reveal your current search path on any LC machine by executing

```
echo $PATH
```

on that machine.

## Compaq-Cluster Search Path

The default search path on every machine in the open and secure Compaq (Tru64) clusters (before you customize PATH) is, in order from the top:

```
/usr/local/bin
/usr/bin
/usr/ccs/bin
/usr/bin/X11
/usr/apps/bin [now a link to /usr/gapps/bin/$SYS_TYPE]
/g/ggroup/uname
.
```

Note these implications of this unusually short search path:

- Starting in June, 2003, /usr/local/bin comes *before* /usr/bin, as is the case on LC's IBM (AIX) and Linux/CHAOS machines as well.
- Your home directory (/g/ggroup/uname or its alias ~) and your current directory (.) are absent from the literal default search path, but the default .login file on the Compaqs appends them AT THE END (as shown here) for practical convenience. Your personal customizations may change this. If you have a private executable file with the same name as some public program, where you keep it will thus determine whether the shell finds your private version before it finds the public version or vice versa.
- The default search path overtly omits the working directory /var/tmp (and its alias /usr/tmp). Thus the usual practice is to move (CD) to that directory before trying to execute files that reside there, since then the dot (.), if added at the end of the search path, will evaluate to /var/tmp.
- On the Compaq (Tru64) machines, directories in the search path are NOT searched recursively. For example, children of /usr/local/apps are NOT searched even if you add this parent directory to your default search path.

- If you want to use any of these shared application programs on Compaq machines:

```
basis
DX
forge
meshtv
mpich
NCAR
parasoft
pvm
yorick
```

you must add an appropriate enabling line to the end of your "domain specific" initialization file(s).  
Follow these steps:

- (1) Go to your home directory.
- (2) Edit .login.compaq (if you use CSH or TCH) or .profile.compaq (if you use SH or KSH).
- (3) At the end of the relevant initialization file, add the appropriate line(s) from this chart for the software you want to use.

For	Add to .login.compaq
---	-----
basis	source /usr/local/etc/basis.setup.csh
DX	source /usr/local/etc/DX.setup.csh
forge	source /usr/local/etc/forge.setup.csh
meshtv	source /usr/local/etc/meshtv.setup.csh
mpich	source /usr/local/etc/mpich.setup.csh
NCAR	source /usr/local/etc/NCAR.setup.csh
parasoft	source /usr/local/etc/parasoft.setup.csh
pvm	source /usr/local/etc/pvm.setup.csh
yorick	source /usr/local/etc/yorick.setup.csh

For	Add to .profile.compaq
---	-----
basis	. /usr/local/etc/basis.setup.sh
DX	. /usr/local/etc/DX.setup.sh
forge	. /usr/local/etc/forge.setup.sh
meshtv	. /usr/local/etc/meshtv.setup.sh
mpich	. /usr/local/etc/mpich.setup.sh
NCAR	. /usr/local/etc/NCAR.setup.sh
parasoft	. /usr/local/etc/parasoft.setup.sh
pvm	. /usr/local/etc/pvm.setup.sh
yorick	. /usr/local/etc/yorick.setup.sh

- (4) If your initialization file exits before the end for batch jobs, add the extra line(s) BEFORE the batch exit occurs.

# Permissions for Files and Directories

## Kinds of Permission

All files and directories have an owner, usually the person who initially created the file or directory. The owner can assign UNIX permissions to other users, and these permissions control who can manipulate the files and directories.

There are three classes of users who may have different permissions for a file or directory:

```
u = user    (the owner)
g = group   (the owner's group)
o = others  (everyone else)
```

There are three kinds of permissions (r, w, x) that may be assigned to any or all of the above classes of users. This table explains each permission and what it allows those who have it to do:

Permission	Actions allowed on files	Actions allowed on directories
R (read)	Read, copy (CAT, CP, LP, etc.)	List files in directory
W (write)	Edit, append	Create or remove files in directory
X (execute)	Run, execute	CD into directory

## How to Discover Permissions

Run LS with the -l option to see the permissions assigned to your files and directories. Use the -a option (ls -la) to also see those files whose names begin with a dot (.). Here is an example of such output:

```
drw----- 1 tom grp 1714 Mar 24 10:34 ABC
-rw----- 1 tom grp 2966 Apr 14 16:48 filea
-rw----- 1 tom grp 64269 Mar 13 15:51 filxyz
-rwxrw-r-- 1 tom grp 309938 Apr 10 11:19 MYFILE
|_____|
|
file permissions
(the very first character is not part of the permissions)
```

The permissions assigned to the file MYFILE in this example can be interpreted as:

rw-	rw-	r--	The 9-character field is divided into 3 ownership classes. The 9 permissions together form the "mode" of the file.
u	g	o	
User has read, write, and execute rights		Others have read but not write or execute rights	
	Group has read, write, no execute rights		

The default permissions for files on all LC machines are RW for the owner and no permissions for any other user (mode rw-----).

To obtain a list of all your files that have certain specific permissions, you need to use not LS but a tricky combination of options for the FIND utility. You also need to use the octal method of representing permissions, explained in the [next section](#) (page 29). As an example of this technique, the execute line

```
find . -perm -770 -follow -print
```

reports all and only the files in your current directory (.) and all of its subdirectories that have access permission 770 (user rwx, group rwx, others none).

# How to Change Permissions (CHMOD)

Only the owner of a file or directory (or the super user) can change a file's permissions (mode). The changes are made with the CHMOD utility. There are two forms of CHMOD syntax: one specifies the desired permissions as an absolute (octal numeric) value; the other is symbolic and changes permissions incrementally.

```
chmod nnn files           (absolute form)
chmod j operator k files   (symbolic form)
```

## Absolute (Octal) Form of CHMOD

In the absolute form of CHMOD a 3-digit octal number *nnn* specifies the desired permissions for a file or directory, without regard for its previous permissions:

```
chmod nnn filename(s)
```

You construct the appropriate enabling number *nnn* by ORing (adding) the digits vertically for each permission you want to allow for each class of user (u, g, o) shown in the chart below:

	u	g	o
r	4	4	4
w	2	2	2
x	1	1	1

For example:	6	4	4	Gives user read and write, everybody else read only
	7	5	5	User has rwx; group and others have rx rights
	6	0	0	User has rw rights, others have no rights

For instance, to assign the permissions (mode) rwxrw-rw- (766) to MYFILE, type the line

```
chmod 766 MYFILE
```

which works regardless of what the former permissions on MYFILE were.

## Symbolic Form of CHMOD

The symbolic version of CHMOD uses letters and symbols to specify incremental changes to the existing permissions on a file (add a right, delete a right, change all the rights) and which user classes receive the change (user, group, others). This chart shows the pattern:

COMMAND	HOW TO ALTER MODE	FILE(S) TO CHANGE
<code>chmod</code>	<code>j operator k</code>	<code>filename</code>

---

where

`j` is    `u` (user)  
          `g` (group)  
          `o` (others)  
          `a` (all of the above)

`operator` is    `-` (remove permission)  
                  `+` (add permission)  
                  `=` (reset/assign permissions;  
                      overrides all current per-  
                      missions)  
                  on all user classes)

`k` is    `r` (read)  
          `w` (write)  
          `x` (execute)

For example, to change the permissions of the file MYFILE from `rw-rw-r--` (user `rw`; group `rw`; others `r`) to `rw-rw-rw-` (user `rw`, group `rw`, others `rw`), you need to add execute permission for the user and write permission for others. So you would type:

```
chmod u+x,o+w MYFILE
```

More than one change may be made, but they must be separated by commas and NO spaces, as shown here. To later remove execute (`x`) permission for the user (with this symbolic CHMOD syntax), type:

```
chmod u-x MYFILE
```

## Citizenship and Permissions (SCF)

On the secure (SCF) production computers (the IBM White, Ice, and SKY nodes and the Compaq clusters), user citizenship can affect file sharing and the assignment of file permissions.

System administrators create groups of users (usually at the request of projects or collaborators) so that group members can use CHGRP and CHMOD to allow shared access to files among, but only among, everyone in the group. The GROUPS utility, run without options, reports the names of the groups to which you currently belong, just as LS reports the group to which a file has been assigned. On the SCF production machines, every user belongs to an "extra" group that reflects that user's citizenship (for example, every U.S. citizen belongs to the group us\_cit). You can therefore restrict the access for any file to fellow citizens by using CHGRP to assign the file to your citizenship group, such as

```
chgrp us_cit myfile
```

and then using CHMOD to open group permissions but limit (or eliminate) world permissions on that file, such as

```
chmod 750 myfile
```

On SCF production machines, system administrators also use these citizenship groups in more subtle ways, along with the segregation of home directories by citizenship, to manage file access generally. If you think your file management activities call for more details on the interaction of citizenship with file permissions, contact the LC hotline at 925-422-4531 with specific questions.

## Top-Level World Permissions Disabled

World (or "other") permissions on top-level files and directories invite unauthorized access and other security problems. Therefore, beginning in April, 1999, an automatic monitoring process systematically disables (converts to permission 0) all world permissions (read, write, and execute) on TOP-level user directories and files in the following file systems

- common-home (/g/gnn ),
- NFS-mounted temporary (on SCF, /nfs/tmp0, /nfs/tmp2; on OCF, /nfs/tmp3),
- /var/tmp (sometimes called /usr/tmp),
- /tmp,
- GPFS (/p/gx1, /p/gx2),
- /usr/gapps (linked from former /usr/apps)

on each LC production machine where they reside (permissions on files below the top level remain unchanged). Access to TOP-level DFS (distributed file system) (page 15) project directories is similarly restricted, but by using the access control list (ACL) mechanism instead of UNIX permissions (see the DFS section above for details).

Because disabling all top-level world permissions (in the specified file systems) is a security policy, exceptions will require a justification memo with the signature of the relevant LLNL Associate Director. Part of that justification must include a statement that the files or directories do not contain, nor will they ever contain, export controlled or sensitive unclassified information. Only system-owned top-level directories (those with UIDs below 1000) avoid this requirement. (Contact the LC Hotline if you want to apply for a specific exemption to the restrictions on world access.)

Several alternative ways to share files safely are available on LC machines, each with its own strengths and weaknesses. See the File Sharing (page 23) section in this guide for a systematic comparison of these alternatives. When planning such safe file sharing, you may want to discover all your files with a specified permission set. Instructions for doing this with FIND appear in the How to Discover Permissions (page 27) section above.



## File-Management Tools

This section summarizes the crucial instructions for several file-management tools that are either unique to the local computing environment (e.g., GIVE and TAKE) or that have special significance here where so many resources are shared by many users (e.g., QUOTA). These software tools are generally available on LC machines unless otherwise noted in the "exceptions" column in the table below. Because you may want to use these tools on several platforms from different vendors or in combination (even though they were written by different people at different times), note also the sometimes-troublesome unit discrepancies in the last column.

<b>Tool:</b>	<b>Function:</b>	<b>Units:</b>
<u>GIVE</u>	Offers files to another user to TAKE, even across machines	
<u>TAKE</u>	Accepts GIVEn files from another user, even across machines	
<u>QUOTA</u>	Reports your current local and global disk usage and limits (includes common home directories)	1024-byte blocks (all Compaqs) mega/gigabytes (all IBMs)
<u>LIMIT</u>	Summarizes machine configuration limits	kbytes
<u>DU</u>	Reports disk usage (for current or specified directory)	512-byte blocks (default) 1024-byte blocks (with -k)
<u>DF</u>	Reports free disk space (for current or specified file system)	512-byte blocks (Compaqs, default) 1024-byte blocks (Compaqs with -k)
<u>MOLE</u>	Displays text files with better control, better status information, better handling of nonstandard content than MORE	lines, characters
<u>HTAR</u>	Bundles files into TAR-format archives or extracts archive members, for archives on <i>remote</i> machines	

NOTE: on LC Linux machines, some standard UNIX file-handling tools have many useful extra options (e.g., DU). All LC-developed file tools have now been ported to Linux.

# GIVE

## How to Run GIVE

GIVE transfers files to another specified user (i.e., changes the owner) on the same machine (sometimes also across machines, see below) by copying the files to a holding directory from which the intended recipient (but no one else) can retrieve the files by running TAKE (page 37). GIVE thus lets any user do what normally only a UNIX superuser can do with CHOWN.

To GIVE a file, type

```
give [-i] [-r] [-l] takename flist
```

```
give -l [takename] | -u takename [flist]
```

where

*takename* is the login name of the user you intend to receive the files. You cannot specify several TAKERs with one execution of GIVE, though you can specify several files.

*flist* is the name, space-delimited list of names, or file filter that specifies the file(s) or directories to GIVE to *takename*. Pathnames are accepted for files not in the current directory.

When you GIVE a file, you always automatically retain the original. Because the transferred copy waits in a special temporary subdirectory it may be purged if the TAKER delays retrieving it longer than the local purge interval (sometimes as short as 10 days). GIVE options let you confirm (-l) or cancel (-u) GIVEN files not yet TAKEN. If you try to GIVE two files with the same name, only the first is copied to the holding directory (you get a warning about the second). GIVEN files always arrive with the TAKER as owner and group, and with 600 or 700 permissions, regardless of how you set the group and permissions on the file before you GIVE it.

On OCF IBM machines (only), GIVEN files are copied to the file system /usr/give, which has a whole-system (not per-user) quota of 10 Gbyte of disk space. If you plan to GIVE a few large files (or even many small ones) on an IBM machine, therefore, you should probably confirm that you will not violate this quota by first running

```
df -k /usr/give
```

to get a current report on how much of that 10 Gbyte is still available free space in /usr/give (see the DF section (page 44) below for more background on using DF). There is no quota on the number of files that you can GIVE.

On both OCF and SCF production machines (but not on some special-purpose machines) and for all three LC operating systems (AIX, Tru64, and Linux/CHAOS), the transfer directory /usr/give is globally mounted. This means that a GIVER can run on one machine and a TAKER can run on another, for across-machine file transfers. Across-machine GIVES and TAKES can transfer files between nodes within one cluster (from one /usr/tmp to another, for example) or between entirely separate, dissimilar clusters (such as Blue and MCR).

## GIVE Options

The -i and -r options control file transfers; the -l and -u options manage files already GIVEN.

-i *takename flist*

causes GIVE to interactively prompt about transferring each file in *flist*, to which you respond y (to GIVE) or n (to omit).

-r *takename flist*

recursively GIVES the contents of all directories in *flist*, including all files in all of their subdirectories. You must use -r if *flist* contains any directories at all.

-l [*takename*]

lists the files you have GIVEN to *takename* that they have not yet retrieved from the holding directory. Without *takename*, lists all unretrieved files you have GIVEN to any user (grouped by user).

-u *takename [flist]*

ungives (withdraws, removes) the specified files (or, by default, all files) that you have GIVEN to *takename* that they have not yet retrieved from the holding directory.

## GIVE Examples

[1]

---

GOAL: To give a file to another user on GPS01, later confirm that it is available for that user to TAKE, then (still later) change your mind and "ungive" the file.

STRATEGY: (1) Run GIVE and specify the user to receive the file (here DMASON) and the file to transfer (here JLOG2). GIVE confirms your donation automatically.  
(2) Later use GIVE's -l option to list your unTAKEn but previously GIVEN files.  
(3) Then use GIVE's -u option to revoke the transfer to DMASON of JLOG2. This ungive is NOT automatically confirmed, but you can check by using the -l option again, as shown.

```
give dmason jlog2          --- (1)
  1 file given.
```

```
give -l                    --- (2)
  dmason has been given:
    1736704 Jan 29 10:30 jlog1

  jsmith has given:
    1640205 Jan 25 11:12 jlog1
```

```
give -u dmason jlog2      --- (3)
  1 files removed.
```

```
give -l  
    You have given 0 files.
```

---

# TAKE

## How to Run TAKE

TAKE transfers files from another specified user (i.e., changes the owner) on the same machine (sometimes also across machines, see below) by copying the files from a holding directory where the donor has previously put copies by running GIVE (page 34) (and indicating you as the intended TAKER). TAKE, together with GIVE, thus lets any user do what normally only a UNIX superuser can do with CHOWN.

To TAKE a file, type

```
take [-i] [-r] givename [flist]
```

```
take [-l [givename]]
```

where

*givename* is the login name of the user who previously ran GIVE to transfer files to you. To actually retrieve any files you must specify the user who gave them, and you cannot TAKE files from several users at once (though you can TAKE several files from one user).

*flist* is the name or space-delimited list of names (but NOT a file filter) that specifies the file(s) or directories to TAKE from *givename*. TAKEn files arrive in the directory where you run TAKE, and for TAKEn directories the whole subdirectory structure will be reproduced in your current directory if that structure does not already exist.

When you TAKE a file (a copy), the GIVER always automatically retains the original. Because the transferred copy waits in a special temporary subdirectory it may be purged if you delay retrieving it longer than the local purge interval (sometimes as short as 10 days). TAKE options let you list (-l) GIVEN files awaiting retrieval or control how files are retrieved (-i, -r). If you try to TAKE a file with the same name as one already in your current directory, TAKE warns you that "xxx already exists locally" and does NOT overwrite the existing file, a safety feature. TAKEn files arrive with you as the owner and group (a contrast with CP or MV) and with default 600 permissions (RW for owner only) or 700 permissions (RWX for owner only), regardless of how the GIVER set the group and permissions on the original file.

On both OCF and SCF production machines (but not on some special-purpose machines) and for all three LC operating systems (AIX, Tru64, and Linux/CHAOS), the transfer directory /usr/give is globally mounted. This means that a GIVER can run on one machine and a TAKER can run on another, for across-machine file transfers. Across-machine GIVES and TAKES can transfer files between nodes within one cluster (from one /usr/tmp to another, for example) or between entirely separate, dissimilar clusters (such as Blue and MCR).

## TAKE Options

The -i and -r options control file transfers; the -l option reports files already GIVEN.

-i *givername* [*flist*]

causes TAKE to interactively prompt about transferring each file in *flist*, to which you respond y (to TAKE) or n (to omit).

-r *givername* [*flist*]

recursively TAKES the contents of all directories in *flist*, including all files in all of their subdirectories. You must use -r if *flist* contains any directories at all.

-l [*givername*]

lists the files that *givername* has already GIVEN you that you have not yet retrieved from the holding directory. Without *givername*, lists all unretrieved files you have been GIVEN by any user (grouped by user). The -l option prevents actual file transfers and just reports available files; typing TAKE or TAKE -l yields the same report.

## TAKE Examples

[1]

---

GOAL: To discover which files have been GIVEN to you by a user on GPS01, and then retrieve a copy of those files.

STRATEGY: (1) Run TAKE with the -l option to list the files you have been GIVEN but have not yet retrieved.  
(2) Then run TAKE with the giver's login name (here DMASON) to actually retrieve the files. You cannot TAKE files without specifying the giver.

```
take -l                                --- (1)
  dmason has given:
    1270 Feb 5 15:04  log2
    7340 Feb 6 09:15  log3
  2 files.

take dmason                            --- (2)
  Taking files.
    2 file(s) taken.
```

---

# QUOTA

## How to Run QUOTA

QUOTA reports your current disk usage both in bytes and in files (sometimes called inodes), as well as the current byte and file limits imposed on you on each file system. In the past, the format of and labels on QUOTA reports varied from one computer vendor or platform to another, and some systems reported disk usage in 1024-byte "blocks." Because those reports were hard to interpret and awkward to display as the numbers grew very large, simple megabyte (M) and gigabyte (G) units now appear in all QUOTA output on LC production machines.

To run QUOTA, type

**quota** [ -v ]

(Note that QUOTA is in /usr/sbin on the Compaqs, so you may need to type /usr/sbin/quota if sbin is not in your search path.)

On open and secure IBM, Linux (Intel), and Compaq production machines, QUOTA with -v reports common-home directory usage and limits directly in megabytes (M) or gigabytes (G). The default report (without -v) is null, and no unit conversion is necessary. Thus, QUOTA with -v now replaces the former GLOBAL\_QUOTA utility for reporting your LC common-home directory usage and limits on all LC production machines. (NOTE: QUOTA also reports the current 1,000,000 files/user limit on all /nfs/tmp file systems on SCF machines.)

## QUOTA Options

Most QUOTA options (not mentioned here) can be used only by system administrators for managing file systems.

-v	On IBM, Linux (Intel), and Compaq machines, -v displays disk usage and limit information on mounted global file systems (common home directories and /nfs/tmp*) only.
----	---

## QUOTA Examples

[1]

---

**GOAL:** To report your current disk usage (including your common-home directory usage) and see how close to the allowed limits you are.

**STRATEGY:** Run QUOTA with the -v option.  
On LC's IBM, Linux (Intel), and Compaq machines, this report is directly in megabytes (M) or gigabytes (G) and covers ONLY the common home directories and the /nfs shared work directories.

```
quota -v
```

```
Disk quotas for jsmith:
```

Filesystem	used	quota	limit	timeleft	files	quota	limit	timeleft
/g/g5	721.7M	8.0G	8.0G		2874	n/a	n/a	
/g/g17	-0-	16.0G	16.0G		0	n/a	n/a	
/g/g50	-0-	16.0G	16.0G		0	n/a	n/a	
[many others too]								
/nfs/tmp3	-0-	100.0G	100.0G		75	1000000	1000000	

---



# LIMIT

## How to Run LIMIT

LIMIT displays a brief summary of the current machine configuration limits. To run LIMIT, just type

```
limit
```

## LIMIT Options

The LIMIT utility has no options.

## LIMIT Examples

[1]

---

GOAL: To report the current machine configuration on GPS01 in the open Compaq cluster.

STRATEGY: Run LIMIT, whose report includes the relevant units for each feature reported.

```
limit

cputime 30:00
filesize unlimited
datasize 3145728 kbytes
stacksize 3145728 kbytes
coredumpsize 0 kbytes
memoryuse 4101400 kbytes
descriptors 4096 files
addressspace 409600 kbytes
```

---

# DU

## How to Run DU

DU ("disk usage") reports the amount of disk space used by a specified directory, and by default, the space used by each of the individual subdirectories of that directory as well. If you have many (sub)directories this can be a helpful aid in planning and monitoring your use of disk space, because DU's report is much more fine-grained than are the summed reports from QUOTA.

To run DU, type

```
du [options] [directory]
```

where

*options* control the units that DU uses and the level of detail in its disk-space reports. By default, DU reports in 512-byte (= 0.5-kbyte) blocks and overtly itemizes all the subdirectories of the target directory.

*directory* specifies (with a pathname) which directory to report on. By default, DU reports on the directory you are in when you run it.

## DU Options

DU's options vary somewhat from one vendor (and hence platform) to another, but these two options are especially helpful and are supported on all local machines.

-k causes DU to report all disk usage in 1024-byte (= 1-kbyte) blocks (the default unit is 512-byte blocks).

-s eliminates the default report of itemized subdirectories and just prints the total disk usage for the one directory you specify on the execute line.

On LC Linux machines only, DU has a dozen useful extra options (mostly to control the block size used or to request totals along with detailed reports).

## DU Examples

[1]

---

**GOAL:** To see how your disk usage is divided among the subdirectories of your home directory (here, on GPS01).

**STRATEGY:** Run DU and specify your home directory (~) as the target for its report. If you use the -k option, as shown here, the report units will be 1-kbyte blocks (the numbers along the left). Note that the total appears at the end of the list, and the children of each subdirectory are listed above it.

```
du -k ~
```

```
900      /g/g0/trg/.dt
180      /g/g0/trg/.elm
4        /g/g0/trg/.netscape/archive
3        /g/g0/trg/Mail
429      /g/g0/trg/anduin/new
434      /g/g0/trg/anduin
704      /g/g0/trg/banks
4032     /g/g0/trg/baxter
164      /g/g0/trg/chome
[...many more subdirectories, alphabetically...]
15       /g/g0/trg/eom
719      /g/g0/trg/fis
5818     /g/g0/trg/mail
1021     /g/g0/trg/nmg
2119     /g/g0/trg/test71a
1144     /g/g0/trg/verify
2        /g/g0/trg/nft/case1/case2/case3
3        /g/g0/trg/nft/case1/case2
1        /g/g0/trg/nft/case1/soft
5        /g/g0/trg/nft/case1
1095     /g/g0/trg/nft
856      /g/g0/trg/sgml2fc
224      /g/g0/trg/sybase
284      /g/g0/trg/jcd
37292    /g/g0/trg
```

---

# DF

## How to Run DF

DF ("disk free") reports the free space left on any mounted device (file systems and their associated directories), both in absolute terms and as a percentage of the total space available to users. To run DF, type

```
df [options] [filesystem]
```

where

- |                   |   |
|-------------------|---|
| <i>options</i>    | control the units that DF uses and the level of detail in its disk-space reports. By default, DF reports in 512-byte (= 0.5-kbyte) blocks on Compaqs and IBMs, and 1024-byte (= 1-kbyte) blocks on Linux (Intel) nodes.   |
| <i>filesystem</i> | specifies (with a pathname) which file system to report on. By default, DF reports on all currently mounted file systems (but it excludes any automounted file systems, such as the "global" file system supporting the common home directories). See <a href="#">QUOTA</a> (page 39) (discussed above) as a supplement. You can use either the physical device name (e.g., /dev/vartmp on IBM nodes or /dev/hda9 on Linux nodes) or the more familiar name of the associated directory (e.g., /var/tmp). |

Details of DF's reports vary by vendor (and hence by platform). Note that when "capacity" (really usage) is reported as a percentage (e.g., 65%), this is not simply a ratio of the used blocks to total blocks (each reported in other columns) because typically some fraction of the total space (often 10%) on a file system is reserved for system use and is not available to users. "Capacity" represents the percentage of this smaller, actually available space that is now *used* (on Linux nodes, this value is more appropriately called "Use%").

## DF Options

DF's options vary somewhat from one vendor (and hence platform) to another, but one especially helpful option is supported on all local machines.

- |    |   |
|----|---|
| -k | causes DF to report all disk usage in 1024-byte (= 1-kbyte) blocks (the default unit is 512-byte blocks on Compaq and IBM nodes). |
|----|---|

## DF Examples

[1]

---

**GOAL:** To see how much user space remains on the working file system supporting /var/tmp (on GPS17 in the open Compaq cluster).

**STRATEGY:** Run DF with the -k option (to get a report in 1-kbyte units) and specify /var/tmp as the target for the report. "Capacity" means "percent used" (and is so reported on IBM and Linux nodes).

```
df -k /var/tmp
```

Filesystem	1024-blocks	Used	Avail	Capacity	Mounted on
utmp_domain#utmp	8123400	4747760	2563300	65%	/var/cluster /members /member0/tmp

---

# MOLE

## How to Run MOLE

MOLE displays text files using MORE-like commands. But MOLE has been enhanced to provide:

- (1) forward *and backward* movement within each file as well as among multiple files,
- (2) status information on file name, size, and creation date (in a header that begins each displayed file) along with current position (in a prompt at the bottom of each screen), and
- (3) counts of hidden lines, signals (bell) for nonASCII characters, and no grotesque attempts to display binary files.

MOLE also lets you jump to a specified line number in the open file, in either direction.

To run MOLE, type

```
mole [flist]
```

where

*flist* is the name, space-delimited list of names, or file filter that specifies the file(s) that you want MOLE to display. Pathnames are accepted for files not in the current directory. If run without *flist*, MOLE displays a 10-line help message and ends.

Once started, MOLE steps through each file in 40-line (default) or 20-line (if you toggle with the \* command) screens, or one line at a time (using RETURN), with optional backward steps (L command) or jumps between files (with + and -). Like MORE, MOLE does not echo typed commands on the screen and does not require a RETURN after each command to execute it. See the MOLE section (URL: <http://www.llnl.gov/LCdocs/fis/index.jsp?show=s6.4>) of the File Interchange Service (FIS) Manual for more details on interpreting MOLE's prompt and file header, and for MOLE's role in helping ADCs review files for secure-to-open transfer.

## MOLE Options

The first four options here MOLE shares with MORE; the others are unique to MOLE.

RETURN	(same as MORE) displays the next line in the open file.
SPACE	(same as MORE) displays the next screen (40 lines by default).
?	(same as MORE) displays a brief help message (command summary).
q	(same as MORE) terminates MOLE (with a report of the number of files reviewed).
*	toggles between whole-screen (40-line, the default) display and half-screen (20-line) display steps.
L	(ell, either case) displays the previous (40-line or 20-line) screen again.
<i>nnn</i> RETURN	displays the line numbered <i>nnn</i> as well as the (40 or 20) lines that immediately precede line <i>nnn</i> (only this option takes a RETURN).

- displays again the previous file (if you have specified several on MOLE's execute line). A one-line header identifies each file displayed when the display starts.
- + displays the (sequentially or alphabetically) next file (if you have specified several on MOLE's execute line). A one-line header identifies each file displayed when the display starts.

## MOLE Examples

[1]

**GOAL:** To display first one file (test1), then another (test2), including jumps to a specific line and display of hidden lines.

**STRATEGY:**

- (1) Run MOLE with both file names on its execute line.
- (2) Toggle to the 20-line step size and request the next 20 lines.
- (3) Jump to line 400 in test1 (and the 20 lines that precede it). As always, MOLE's prompt reveals this new position by line number.
- (4) Move to the "next" file (test2). Note that MOLE reports in its one-line file header that test2 contains 8 hidden lines, and any that fall near the front will automatically appear along with the first 20 lines displayed here.
- (5) Quit MOLE (note the final report).

User: mole test1 test2	---	(1)
Rtne: [test1]: 408L/21957 Jan 5, 1998 11:03	[MOLE's header]	
...first 40 lines of test1 displayed here...	[text]	
? l/f space L # + - * Q *** 40/A test1 ***	[MOLE's prompt]	
User: *	---	(2)
Rtne: [bell]	[MOLE acknowledges]	
User: [space]	[request next screen]	
Rtne:		
...next 20 lines of test1 displayed here...	[text]	
? l/f space L # + - * Q *** 60/A test1 ***	[MOLE's prompt]	
User: 400RETURN	---	(3)
Rtne:		
...20 lines of test1 ending at line 400...		
? l/f space L # + - * Q *** 400/A test1 ***	[note new location]	

```
User: +                                ---(4)
Rtne: [test2]: 110L/5331 Apr 9, 2001 3:17 *** 8 hidden lines***
                                           [MOLE's header]
      ...first 20 lines of test2 displayed here, [text]
      including any hidden lines in this range...
      ? l/f space L # + - * Q *** 20/A test2 *** [MOLE's prompt]
User: q                                ---(5)
Rtne: [2 ASCII files reviewed]          [final status report]
```

---



# HTAR

HTAR is basically a (locally developed) interface to LC's (open or secure) storage system HPSS. But it behaves like (and shares much execute-line syntax with) the traditional UNIX TAR utility, so in that sense HTAR is also a file-bundling and archive-management tool. Also (like LC's NFT (URL: <http://www.llnl.gov/LCdocs/nft>)), you can request HTAR to interact with remote archives on *any* LC machine if you wish (using -F).

HTAR's specially tuned features include:

- Uses a TAR-like syntax and supports TAR-compatible archive files by relying on the POSIX 1003.1 TAR file format.
- Bundles files in memory using multiple concurrent threads and transfers them into an archive file built *directly* in storage by default (or on any *remote* LC machine that you specify), to avoid needing extra local online disk space.
- Takes advantage of available parallel interfaces to storage (PFTP) to provide fast file transfers (measured at as high as 150 Mbyte/s, which exceeds 30 times the typical rate for transferring small files separately).
- Uses an external index file to easily accommodate thousands of small files in any archive, and to support retrieval of specified files from within a remote archive *without* first retrieving the whole archive from HPSS (or from another remote machine).
- Imposes no limit on the total size of the archives that it builds and accepts input files (archive members) as large as 8 Gbyte.

When the storage system (HPSS) is up and available to users you can execute HTAR with a command line that has the general form

```
htar action archive [options] [filelist]
```

and the specific form

```
htar -c|t|x|X -f archivename [-BdEFhHILmMopSTvVw] [flist]
```

where exactly one action and the *archivename* are always required, while the control options and (except when using -c) the *filelist* can be omitted (and the options can share a hyphen flag with the action for convenience). Users familiar with TAR can guess how to run HTAR from this model (although there are some tricky syntax differences). Others should consult the HTAR Reference Manual (URL: <http://www.llnl.gov/LCdocs/htar>) for usage suggestions, annotated examples, technical tips, full option details, and known problems.

## Tools for Obsolete File Types

To help you properly identify still-needed files that were made on LC's former CRAY and CDC 7600 computers, and to (optionally) convert some files that can be rescued, LC provides several tools for managing obsolete file types. These special tools usually perform familiar tasks (such as listing file formats or unpacking libraries) but on file types ignored by standard UNIX tools. See also the "Cray File Conversion" section of the EZOUTPUT (URL: <http://www.llnl.gov/LCdocs/ezoutput>) basic guide for how IBM's (but *not* Compaq's) TRANS tool can sometimes help with such legacy files.

Among the available obsolete-file tools on LC production machines are:

`lft [-c|-n] filelist`

reports for each file in *filelist* (a blank delimited list of files or a UNIX file filter) two columns that specify each file's name and its file type (ascii, directory, lib, tar, unrecognized). LFT reports GIF, PDF, and PS files as ascii. With -n, LFT reports on CRAY and CDC 7600 files without conversion. With -c, LFT converts each (specified) CRAY or CDC 7600 text file to UNIX format (but leaves all other files unchanged).

`lib76 libname l|x|b|a [filelist]`

processes CDC 7600 LIB-format and LIX-format library files. Specify the library with *libname* and the files to process within it with *filelist*. Here option l lists the files, x extracts them as text, b extracts them as binary files, and a extracts them as ascii and converts them to UNIX format. With no *filelist*, LIB76 processes all files in the specified library. WARNING: odd-number length files will gain 4 extra bits, and LIB76 will overwrite without warning any existing local files with the same name as those it extracts.

# Using Groups

## BASIC ADVICE:

A group is a named set of users created by a system administrator to enable easier file sharing among group members. Every user at LC belongs by default to a group (of one) with the same name as their login name, and your newly created files are assigned by default to that unique group. But if you belong to other groups as well, you can change your default group (for the current session) or change the group to which any of your files is assigned, to take advantage of the group permissions (page 27) explained in a previous section. (On LC machines, software constraints limit every user to membership in no more than 32 groups.)

This table shows how to perform the most common group-related tasks:

Group-related task:	Command:
Reveal who belongs to a specified group	<code>uinfo group <i>grpname</i></code> <code>grep <i>grpname</i> /etc/group</code>
Reveal all groups to which you belong	<code>groups</code> <code>uinfo user <i>username</i></code>
Reveal all groups to which <i>username</i> belongs	<code>groups <i>username</i></code> <code>uinfo user <i>username</i></code>
Change your default group to <i>grpname</i>	<code>newgrp <i>grpname</i></code>
Restore your original default group	<code>newgrp</code>
Change a file's group assignment	<code>chgrp <i>grpname filename</i></code>
Change a file's group permissions	See <u><a href="#">chmod</a></u>
Create or join a group at LC	Contact <u><a href="#">LC Hotline</a></u>

## WARNINGS:

(1) Storage Groups. Your group membership on LC's archival storage systems (storage.llnl.gov and storage.scf.cln) is likely to differ from your group membership on LC's production machines (which also often differ among themselves). You can use group assignments to control the sharing of stored files, but only if you discover who belongs to which storage groups, and only if you assign a file to a group AFTER you store it (since group assignment does NOT persist during file transfer). To reveal your group memberships (or the members of a specified group) on the LC storage systems you must run the DCECP utility as explained in the "Using Storage Groups" section of the [EZSTORAGE](#) (URL: <http://www.llnl.gov/LCdocs/ezstorage>) guide.

You can still change the group to which a stored file is assigned by using NFT's **chgrp** option or FTP's **quote site chgrp** option, but only if you belong to the target group. (Despite its name, NFT's **group** option begins asynchronous file transfers and has nothing to do with managing file-permission groups.)

(2) DFS Groups. LC's Distributed File System (DFS) handles groups differently than the UNIX operating system does. As with storage, a file's previous mainframe group assignment reverts to your default group (the one-member group that matches your login name) when you move or copy a file into DFS. Under DFS, a file can be assigned to many groups (not just one as with UNIX), all posted by the **aclmod** utility on the file's access control list (ACL). If you belong to several groups, your DFS permissions for a file are the logical sum (union) of the permissions of every group you belong to that is assigned to the file. And

finally, under DFS you can use **aclmod** to assign a group to a file even if you do not belong to that group yourself (in contrast with **chgrp**). Because of these differences, you should carefully review the instructions for and examples of **aclmod** in LC's DFS documentation before you rely on group behavior under DFS. (Note that DFS is mounted on LC IBM nodes and some Compaq nodes (such as GPS and SC), but *not* on any Linux nodes or on LC's special-purpose Suns.)

#### RELEVANT FORMS:

To create, delete, or change the membership of a group on any open or SCF machines, you must submit one of these (signed) forms to the LC Hotline (L-63).

SCF-2 Create/Update Group  
SCF-9 Delete Group

The blank forms are available by request from the Hotline (page 00) and from these LC web sites:

Open: <https://www.llnl.gov/lcforms>  
SCF: <https://www.scf.cln/lcforms>

The section once linked from this reference no longer exists.

# Disclaimer

---

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.

(C) Copyright 2003 The Regents of the University of California. All rights reserved.

---

# Keyword Index

To see an alphabetical list of keywords for this document, consult the next section (page 57).

Keyword	Description
<u>entire</u>	This entire document.
<u>title</u>	The name of this document.
<u>scope</u>	Topics covered in EZFILES.
<u>availability</u>	Where these programs run.
<u>who</u>	Who to contact for assistance.
<u>introduction</u>	Role and goals of EZFILES.
<u>directories</u>	How public directories are organized.
<u>dec-directories</u>	Directories, properties on open clusters.
<u>dec-structure</u>	How open cluster directories are organized.
<u>dec-properties</u>	Home, work dirs. compared on clusters.
<u>dec-scf-directories</u>	Directories, properties on secure clusters.
<u>dec-structure-scf</u>	How SCF cluster directories are organized.
<u>dec-properties-scf</u>	Home, work dirs. compared on SCF clusters.
<u>ibm-directories</u>	Directories, properties on IBM SPs.
<u>pentium-directories</u>	Directories, properties of Pentium clusters.
<u>dfs-nfs-comparison</u>	Contrasts DFS and NFS shared disks.
<u>nfs</u>	Network File System properties.
<u>dfs</u>	Distributed File System properties.
<u>common-home</u>	Home directories shared across machines.
<u>file-management</u>	Techniques and issues in work with files.
<u>work-steps</u>	Typical code-development scenario.
<u>transfer-rates</u>	CP vs. FTP from mounted file systems.
<u>suitability</u>	Intended-use comparison of file systems.
<u>file-sharing</u>	File-sharing alternatives compared.
<u>backup</u>	LC file backup policy by directory.
<u>search-paths</u>	Search path features analyzed.
<u>dec-path</u>	Annotated cluster default search path.
<u>permissions</u>	Managing UNIX file permissions.
<u>access-rights</u>	Managing UNIX file permissions.
<u>permission-kinds</u>	R, W, X for U, G, O explained.
<u>permission-reports</u>	How to discover file permissions.
<u>chmod</u>	How to change file permissions.
<u>chmod-octal</u>	Absolute, numeric CHMOD syntax.
<u>chmod-symbolic</u>	Symbolic, incremental CHMOD syntax.
<u>citizenship-groups</u>	How citizenship affects SCF permissions.
<u>permission-disabled</u>	Where top-level world perms disabled.
<u>file-tools</u>	Local software tools for file handling.
<u>give</u>	Donating files to another user.
<u>give-execute-line</u>	How to run GIVE.
<u>give-options</u>	Options to control GIVE use.
<u>give-examples</u>	Sample annotated uses of GIVE.
<u>take</u>	Receiving files from another user.

<a href="#"><u>take-execute-line</u></a>	How to run TAKE.
<a href="#"><u>take-options</u></a>	Options to control TAKE use.
<a href="#"><u>take-examples</u></a>	Sample annotated uses of TAKE.
<a href="#"><u>quota</u></a>	Reporting your max. limits on file sys.
<a href="#"><u>quota-execute-line</u></a>	How to run QUOTA.
<a href="#"><u>quota-options</u></a>	Options to control QUOTA use.
<a href="#"><u>quota-examples</u></a>	Sample annotated uses of QUOTA.
<a href="#"><u>limit</u></a>	Reporting machine configuration limits.
<a href="#"><u>limit-execute-line</u></a>	How to run LIMIT.
<a href="#"><u>limit-options</u></a>	Options to control LIMIT use.
<a href="#"><u>limit-examples</u></a>	Sample annotated uses of LIMIT.
<a href="#"><u>du</u></a>	Reporting disk usage by (sub)directory.
<a href="#"><u>du-execute-line</u></a>	How to run DU.
<a href="#"><u>du-options</u></a>	Options to control DU use.
<a href="#"><u>du-examples</u></a>	Sample annotated uses of DU.
<a href="#"><u>df</u></a>	Reporting free space by file system.
<a href="#"><u>df-execute-line</u></a>	How to run DF.
<a href="#"><u>df-options</u></a>	Options to control DF use.
<a href="#"><u>df-examples</u></a>	Sample annotated uses of DF.
<a href="#"><u>mole</u></a>	Displaying text files reliably.
<a href="#"><u>mole-execute-line</u></a>	How to run MOLE.
<a href="#"><u>mole-options</u></a>	Options to control MOLE use.
<a href="#"><u>mole-examples</u></a>	Sample annotated uses of MOLE.
<a href="#"><u>htar</u></a>	Making, managing file archives.
<a href="#"><u>lft</u></a>	Tool for obsolete (legacy) file types.
<a href="#"><u>lib76</u></a>	Tool for obsolete (legacy) file types.
<a href="#"><u>groups</u></a>	Revealing or changing file groups.
<a href="#"><u>index</u></a>	The structural index of keywords.
<a href="#"><u>a</u></a>	The alphabetical index of keywords.
<a href="#"><u>date</u></a>	The latest changes to EZFILES.
<a href="#"><u>revisions</u></a>	The complete revision history.



# Alphabetical List of Keywords

Keyword -----	Description -----
<u>a</u>	The alphabetical index of keywords.
<u>access-rights</u>	Managing UNIX file permissions.
<u>availability</u>	Where these programs run.
<u>backup</u>	LC file backup policy by directory.
<u>chmod</u>	How to change file permissions.
<u>chmod-octal</u>	Absolute, numeric CHMOD syntax.
<u>chmod-symbolic</u>	Symbolic, incremental CHMOD syntax.
<u>citizenship-groups</u>	How citizenship affects SCF permissions.
<u>common-home</u>	Home directories shared across machines.
<u>date</u>	The latest changes to EZFILES.
<u>dec-directories</u>	Directories, properties on open clusters.
<u>dec-path</u>	Annotated cluster default search path.
<u>dec-properties</u>	Home, work dirs. compared on clusters.
<u>dec-properties-scf</u>	Home, work dirs. compared on SCF clusters.
<u>dec-scf-directories</u>	Directories, properties on secure clusters.
<u>dec-structure</u>	How cluster directories are organized.
<u>dec-structure-scf</u>	How SCF cluster directories are organized.
<u>df</u>	Reporting free space by file system.
<u>df-examples</u>	Sample annotated uses of DF.
<u>df-execute-line</u>	How to run DF.
<u>df-options</u>	Options to control DF use.
<u>dfs</u>	Distributed File System properties.
<u>dfs-nfs-comparison</u>	Contrasts DFS and NFS shared disks.
<u>directories</u>	How public directories are organized.
<u>du</u>	Reporting disk usage by (sub)directory.
<u>du-examples</u>	Sample annotated uses of DU.
<u>du-execute-line</u>	How to run DU.
<u>du-options</u>	Options to control DU use.
<u>entire</u>	This entire document.
<u>file-management</u>	Techniques and issues in work with files.
<u>file-tools</u>	Local software tools for file handling.
<u>file-sharing</u>	File-sharing alternatives compared.
<u>give</u>	Donating files to another user.
<u>give-examples</u>	Sample annotated uses of GIVE.
<u>give-execute-line</u>	How to run GIVE.
<u>give-options</u>	Options to control GIVE use.
<u>groups</u>	Revealing or changing file groups.
<u>htar</u>	Making, managing file archives.
<u>ibm-directories</u>	Directories, properties on IBM SPs.
<u>index</u>	The structural index of keywords.
<u>introduction</u>	Role and goals of EZFILES.
<u>lib76</u>	Tool for obsolete (legacy) file types.
<u>limit</u>	Reporting machine configuration limits.
<u>limit-examples</u>	Sample annotated uses of LIMIT.
<u>limit-execute-line</u>	How to run LIMIT.
<u>limit-options</u>	Options to control LIMIT use.
<u>lft</u>	Tool for obsolete (legacy) file types.
<u>mole</u>	Displaying text files reliably.
<u>mole-examples</u>	Sample annotated uses of MOLE.
<u>mole-execute-line</u>	How to run MOLE.
<u>mole-options</u>	Options to control MOLE use.

<u>nfs</u>	Network File System properties.
<u>pentium-directories</u>	Directories, properties of Pentium clusters.
<u>permission-disabled</u>	Where top-level world perms disabled.
<u>permission-kinds</u>	R, W, X for U, G, O explained.
<u>permission-reports</u>	How to discover file permissions.
<u>permissions</u>	Managing UNIX file permissions.
<u>quota</u>	Reporting your max. limits on file sys.
<u>quota-examples</u>	Sample annotated uses of QUOTA.
<u>quota-execute-line</u>	How to run QUOTA.
<u>quota-options</u>	Options to control QUOTA use.
<u>revisions</u>	The complete revision history.
<u>scope</u>	Topics covered in EZFILES.
<u>search-paths</u>	Search path features analyzed.
<u>suitability</u>	Intended-use comparison of file systems.
<u>take</u>	Receiving files from another user.
<u>take-examples</u>	Sample annotated uses of TAKE.
<u>take-execute-line</u>	How to run TAKE.
<u>take-options</u>	Options to control TAKE use.
<u>title</u>	The name of this document.
<u>transfer-rates</u>	CP vs. FTP from mounted file systems.
<u>who</u>	Who to contact for assistance.
<u>work-steps</u>	Typical code-development scenario.

## Date and Revisions

Revision Date -----	Keyword Affected -----	Description of Change -----
08Dec03	<u>pentium-directories</u>	ACE SCF details added.
	<u>file-tools</u>	GIVE, TAKE now cross all machines.
	<u>give</u>	General across-machine GIVES ok.
	<u>take</u>	General across-machine TAKES ok.
01Oct03	<u>dec-directories</u>	LX cluster departs.
	<u>dec-scf-directories</u>	Furnace departs, ICF noted.
	<u>pentium-directories</u>	MCR and other details added.
	<u>common-home</u>	Furnace departs, MCR added.
08Sep03	<u>common-home</u>	Larger SCF home directories.
	<u>quota</u>	QUOTA output now uniform everywhere.
04Aug03	<u>directories</u>	/nfs/tmp subnames, numbers updated.
	<u>work-steps</u>	/nfs/tmp3 added.
	<u>backup</u>	/nfs/tmp3 added.
	<u>permission-disabled</u>	/nfs/tmp3 added.
	<u>htar</u>	Between-machines role (-F) added.
09Jul03	<u>dec-directories</u>	/nfs/tmp* purge clarified.
	<u>dec-scf-directories</u>	/nfs/tmp* purge clarified.
	<u>pentium-directories</u>	More ILX nodes added.
09Jun03	<u>common-home</u>	How to test for availability.
	<u>htar</u>	Now available on Linux too.
	<u>give</u>	Some across-machine GIVES ok.
	<u>take</u>	Some across-machine TAKES ok.
	<u>search-paths</u>	Tru64 order changed, ECHO noted.
16Apr03	<u>common-home</u>	Name changes, new OCF quotas.
	<u>backup</u>	Clarifying cross ref added.
	<u>quota</u>	Col heads, default output elaborated.
	<u>df</u>	Between-machine differences clarified.
24Mar03	<u>introduction</u>	Cross ref to CHAOS manual added.
	<u>dec-directories</u>	Machine names, details updated.
	<u>dec-scf-directories</u>	LC/Furnace match noted.
	<u>pentium-directories</u>	ILX cluster added, with CHAOS links.
	<u>transfer-rates</u>	ILX replaces LX in example.
	<u>suitability</u>	On Linux, /tmp links to /var/tmp.
	<u>file-tools</u>	Local tools now on Linux too.

01Oct02	<u>dec-scf-directories</u>	Clarified because Forest departs.
	<u>pentium-directories</u>	DFS not on any Linux nodes.
	<u>dfs</u>	DFS not on any Linux nodes.
	<u>backup</u>	DFS has auto backup.
	<u>groups</u>	DFS access limits noted.
29Jul02	<u>ibm-directories</u>	GPFS purge details added.
	<u>work-steps</u>	IBM /var/tmp subdirs noted.
01Jul02	<u>lft</u>	New section on legacy-support tool.
	<u>lib76</u>	New section on legacy-support tool.
	<u>groups</u>	UINFO roles added.
	<u>index</u>	New keywords for new section.
21May02	<u>dec-scf-directories</u>	/var/tmp size variations noted.
	<u>pentium-directories</u>	New section on SCF Pentium clusters.
	<u>common-home</u>	Linux dot file added.
	<u>index</u>	New keyword for new section.
05Mar02	<u>directories</u>	New clusters replace old ones (Compass out, GPS in, etc.).
	<u>common-home</u>	Example and details updated.
	<u>file-tools</u>	Many example details revised.
	<u>index</u>	Separate TERA section deleted.
04Dec01	<u>dec-scf-directories</u>	Home /g replaces /u on Forest.
	<u>ibm-directories</u>	Purge started on SCF GPFS.
	<u>file-tools</u>	Some LC tools not on Linux. DU has extra Linux options.
01Oct01	<u>give</u>	IBM-only GIVE quota explained.
06Sep01	<u>file-tools</u>	Table updated re HTAR.
	<u>htar</u>	New section on new local tool.
	<u>index</u>	New keyword for new section.
17Jul01	<u>dec-scf-directories</u>	SCF /nfs/tmp1 links to /nfs/tmp2.
	<u>ibm-directories</u>	SCF /nfs/tmp1 links to /nfs/tmp2.
	<u>dfs-nfs-comparison</u>	SCF /nfs/tmp1 links to /nfs/tmp2.
	<u>work-steps</u>	SCf /nfs/tmp1 links to /nfs/tmp2.
	<u>groups</u>	Now 32 groups per user.
04Jun01	<u>dec-properties</u>	Limit on /nfs/tmp files/user added.
	<u>dec-properties-scf</u>	Limit on /nfs/tmp files/user added.
	<u>ibm-directories</u>	Limit on /nfs/tmp files/user added.
	<u>quota</u>	/nfs/tmp limit not reported.

17May01	<u>mole</u> <u>index</u>	New section on new local tool. New keywords for new subsections.
03May01	entire	/usr/gapps replaces /usr/apps (OCF, SCF). /nfs/tmp0 and tmp1 replace tmp2 (OCF).
12Mar01	<u>dfs</u> <u>permission-disabled</u>	DFS access restrictions explained. Cross ref to DFS access limits.
04Dec00	<u>directories</u> <u>suitability</u> <u>work-steps</u> <u>backup</u> <u>permission-disabled</u>	/nfs/tmp2 replaces /nfs/tmp1 on OCF. /nfs/tmp2 added to comparison. /nfs/tmp2 replaces /nfs/tmp1 on OCF. /nfs/tmp2 not backed up. /nfs/tmp2 added to list.
14Nov00	<u>search-paths</u>	New DEC/Compaq search path. How to add application setup files.
21Sep00	<u>file-sharing</u> <u>suitability</u> <u>ibm-directories</u> <u>groups</u> <u>backup</u> <u>index</u>	New section compares ways to share files. New section compares file-system roles. New section on SPs, GPFS. Only 16 groups per user. GPFS not backed up. New keywords for new sections.
08Mar00	entire	All CRAY references deleted, including sects on CRAY dirs, search paths.
14Dec99	<u>dec-scf-directories</u> <u>j90-directories</u> <u>common-home</u> <u>dfs-nfs-comparison</u>	/g now optional SCF common home. /g now optional SCF common home. /g mostly replaces /u (SCF). New URL for DFS details.
12Aug99	<u>dec-structure-scf</u> <u>dec-properties-scf</u>	Added /nfs/tmp1 to diagram. Added /nfs/tmp1 col to table.
20Jul99	<u>dec-scf-directories</u> <u>index</u>	Two subsections added. New keywords for new sections.
11May99	<u>permission-reports</u> <u>permission-disabled</u> <u>index</u>	How FIND supplements LS. World permission restrictions. New keyword for new section.
25Mar99	<u>introduction</u> <u>common-home</u> <u>file-tools</u> <u>quota</u> <u>index</u>	GLOBAL_QUOTA deleted. SCF clarified, GLOBAL_QUOTA gone. Table updated re QUOTA. GLOBAL_QUOTA sections deleted. Role expanded, examples revised. All GLOBAL_QUOTA keywords deleted.

20Jan99	<u>common-home</u>	/gadmin/etc replaces /g/etc.
15Dec98	<u>common-home</u>	World write access forbidden.
18Nov98	<u>dec-directories</u>	Shared /nfs/tmp1 dir. explained.
	<u>dec-scf-directories</u>	Absence of /nfs/tmp1 noted.
	<u>file-management</u>	Role of /nfs/tmp1 included.
	<u>dfs-nfs-comparison</u>	New section, details added.
	<u>index</u>	New keywords for new section.
18Sep98	<u>groups</u>	Storage group info updated.
	<u>quota</u>	New role on open DEC's.
	<u>global-quota</u>	Now SCF use only.
	<u>give</u>	Permissions, groups clarified
	<u>take</u>	Permissions, groups clarified
	<u>file-tools</u>	Comparative chart updated.
	<u>common-home</u>	Backup added, details updated.
13Aug98	<u>groups</u>	Groups advice section added.
22Jun98	<u>tera-cluster</u>	New section on new machines.
	<u>citizenship-groups</u>	SCF citizenship effects noted.
	<u>backup</u>	Y-MP, migration comments deleted.
	<u>index</u>	New keywords for new sections.
	<u>entire</u>	Y-MP details deleted throughout.
04Feb98	<u>dec-directories</u>	/u deleted, details clarified.
	<u>dec-scf-directories</u>	New section on Forest added.
	<u>common-home</u>	Forest SCF comments added.
	<u>work-steps</u>	Open/SCF clarifications.
	<u>dec-path</u>	Open/SCF clarifications.
	<u>global-quota</u>	Forest SCF quotas noted.
	<u>index</u>	New keyword for new section.
06Jan98	<u>dec-properties</u>	Size quotas updated.
	<u>who</u>	Link/ref to doc matrix added.
28Oct97	<u>global-quota</u>	File quota gone, DU note added.
	<u>dec-properties</u>	File (count) quota gone.
03Oct97	<u>global-quota</u>	Usage details, example added.
	<u>common-home</u>	Link, server names updated.
	<u>dec-properties</u>	Home quotas, servers updated.
04Jun97	<u>global-quota</u>	Initial use on SKY noted.
18Feb97	<u>entire</u>	First edition of LC EZFILES manual.

TRG (08Dec03)

UCRL-WEB-200067

Privacy and Legal Notice (URL: <http://www.llnl.gov/disclaimer.html>)

TRG (08Dec03) Contact: [lc-hotline@llnl.gov](mailto:lc-hotline@llnl.gov)